SPIDER-DAY05

1. selenium爬虫

1.1 selenium概述

```
【1】定义
    1.1) 开源的Web自动化测试工具

【2】用途
    2.1) 对Web系统进行功能性测试,版本迭代时避免重复劳动
    2.2) 兼容性测试(测试web程序在不同操作系统和不同浏览器中是否运行正常)
    2.3) 对web系统进行大数量测试

【3】特点
    3.1) 可根据指令操控浏览器
    3.2) 只是工具，必须与第三方浏览器结合使用

【4】安装
    4.1) Linux: sudo pip3 install selenium
    4.2) Windows: python -m pip install selenium
```

1.2 PhantomJS概述

```
phantomjs为无界面浏览器(又称无头浏览器)，在内存中进行页面加载,高效
```

1.3 环境安装

- **环境安装**

```
【1】环境说明（'以下三种环境任意安装其中一种即可'）
    环境一：selenium + PhantomJS
    环境二：selenium + chromedriver + Chrome （'我们安装此版本'）
    环境三：selenium + geckodriver + Firefox

【2】提前下载驱动
```

```
    2.1) chromedriver : 下载对应版本
        http://npm.taobao.org/mirrors/chromedriver/
    2.2) geckodriver
        https://github.com/mozilla/geckodriver/releases
    2.3) phantomjs
        https://phantomjs.org/download.html

【3】添加到系统环境变量
    2.1) windows: 将解压后的可执行文件拷贝到Python安装目录的Scripts目录中
         windows查看python安装目录(cmd命令行): where python
    2.2) Linux :  将解压后的文件拷贝到/usr/bin目录中
         sudo cp chromedriver /usr/bin/

【3】Linux中需要修改权限
    sudo chmod 777 /usr/bin/chromedriver

【4】验证
    4.1) Ubuntu | Windows
        from selenium import webdriver
        webdriver.Chrome()
        webdriver.Firefox()

    4.2) Mac
        from selenium import webdriver
        webdriver.Chrome(executable_path='/Users/xxx/chromedriver')
        webdriver.Firefox(executable_path='/User/xxx/geckodriver')
```

2. selenium详解

2.1 代码演示

```python
"""示例代码一: 使用 selenium+浏览器 打开百度"""

# 导入seleinum的webdriver接口
from selenium import webdriver
import time

# 创建浏览器对象
driver = webdriver.Chrome()
driver.get('http://www.baidu.com/')
# 5秒钟后关闭浏览器
time.sleep(5)
driver.quit()
```

```python
"""示例代码二：打开百度，搜索赵丽颖，点击搜索，查看"""

from selenium import webdriver
import time

# 1.创建浏览器对象 - 已经打开了浏览器
driver = webdriver.Chrome()
# 2.输入：http://www.baidu.com/
driver.get('http://www.baidu.com/')
# 3.找到搜索框,向这个节点发送文字：赵丽颖
driver.find_element_by_xpath('//*[@id="kw"]').send_keys('赵丽颖')
# 4.找到 百度一下 按钮,点击一下
driver.find_element_by_xpath('//*[@id="su"]').click()
```

2.2 浏览器对象方法

```
【1】driver.get(url=url)    - 地址栏输入url地址并确认
【2】driver.quit()          - 关闭浏览器
【3】driver.close()         - 关闭当前页
【4】driver.page_source     - HTML结构源码
【5】driver.page_source.find('字符串')
    从html源码中搜索指定字符串,没有找到返回：-1,经常用于判断是否为最后一页
【6】driver.maximize_window() - 浏览器窗口最大化
```

2.3 定位节点

```
【1】单元素查找('结果为1个节点对象')
    1.1) 【最常用】driver.find_element_by_id('id属性值')
    1.2) 【最常用】driver.find_element_by_name('name属性值')
    1.3) 【最常用】driver.find_element_by_class_name('class属性值')
    1.4) 【最万能】driver.find_element_by_xpath('xpath表达式')
    1.5) 【匹配a节点时常用】driver.find_element_by_link_text('链接文本')
    1.6) 【匹配a节点时常用】driver.find_element_by_partical_link_text('部分链接文本')
    1.7) 【最没用】driver.find_element_by_tag_name('标记名称')
    1.8) 【较常用】driver.find_element_by_css_selector('css表达式')

【2】多元素查找('结果为[节点对象列表]')
    2.1) driver.find_elements_by_id('id属性值')
    2.2) driver.find_elements_by_name('name属性值')
    2.3) driver.find_elements_by_class_name('class属性值')
    2.4) driver.find_elements_by_xpath('xpath表达式')
    2.5) driver.find_elements_by_link_text('链接文本')
    2.6) driver.find_elements_by_partical_link_text('部分链接文本')
    2.7) driver.find_elements_by_tag_name('标记名称')
    2.8) driver.find_elements_by_css_selector('css表达式')
```

2.4 节点对象操作

```
【1】node.send_keys('')  - 向文本框发送内容
【2】node.click()       - 点击
【3】node.get_attribute('属性名')  -  获取节点的属性值
【4】node.text  -  获取当前节点及子节点和后代节点的文本内容
```

2.5 猫眼电影爬虫

```
"""
抓取猫眼电影top100的 电影名称、主演、上映时间
URL地址: https://maoyan.com/board/4
"""
from selenium import webdriver
import time

url = 'https://maoyan.com/board/4'
driver = webdriver.Chrome()
driver.get(url)

def get_data():
    # 基准xpath: [<selenium xxx li at xxx>,<selenium xxx li at>]
    li_list = driver.find_elements_by_xpath('//*[@id="app"]/div/div/div[1]/dl/dd')
    for li in li_list:
        item = {}
        # info_list: ['1', '霸王别姬', '主演: 张国荣', '上映时间: 1993-01-01', '9.5']
        info_list = li.text.split('\n')
        item['number'] = info_list[0]
        item['name'] = info_list[1]
        item['star'] = info_list[2]
        item['time'] = info_list[3]
        item['score'] = info_list[4]

        print(item)

while True:
    get_data()
    try:
        driver.find_element_by_link_text('下一页').click()
        time.sleep(2)
    except Exception as e:
        print('恭喜你!抓取结束')
        driver.quit()
        break
```

3. selenium高级

3.1 设置无界面模式

```
from selenium import webdriver

options = webdriver.ChromeOptions()
# 添加无界面参数
options.add_argument('--headless')
driver = webdriver.Chrome(options=options)
```

3.2 鼠标操作

```
"""
鼠标操作三步走:
1、创建鼠标事件类对象
2、指定鼠标行为
3、执行
"""
from selenium import webdriver
# 导入鼠标事件类
from selenium.webdriver import ActionChains

driver = webdriver.Chrome()
driver.get('http://www.baidu.com/')

# 移动到 设置，perform()是真正执行操作，必须有
element = driver.find_element_by_xpath('//*[@id="u1"]/a[8]')
ActionChains(driver).move_to_element(element).perform()

# 单击，弹出的Ajax元素，根据链接节点的文本内容查找
driver.find_element_by_link_text('高级搜索').click()
```

3.3 切换页面

- **适用网站+应对方案**

```
【1】适用网站类型
    页面中点开链接出现新的窗口，但是浏览器对象driver还是之前页面的对象，需要切换到不同的窗口进行操作

【2】应对方案 - driver.switch_to.window()

    # 获取当前所有句柄（窗口）- [handle1,handle2]
    all_handles = driver.window_handles
    # 切换browser到新的窗口，获取新窗口的对象
    driver.switch_to.window(all_handles[1])
```

- **民政部网站案例-selenium**

```
"""
适用selenium+Chrome抓取民政部行政区划代码
http://www.mca.gov.cn/article/sj/xzqh/2020/
"""
from selenium import webdriver

class GovSpider(object):
    def __init__(self):
        # 设置无界面
        options = webdriver.ChromeOptions()
        options.add_argument('--headless')
        # 添加参数
        self.driver = webdriver.Chrome(options=options)
        self.one_url = 'http://www.mca.gov.cn/article/sj/xzqh/2019/'

    def get_incr_url(self):
        self.driver.get(self.one_url)
        # 提取最新链接节点对象并点击

 self.driver.find_element_by_xpath('//td[@class="arlisttd"]/a[contains(@title,"代码")]').click()
        # 切换句柄
        all_handlers = self.driver.window_handles
        self.driver.switch_to.window(all_handlers[1])
        self.get_data()

    def get_data(self):
        tr_list = self.driver.find_elements_by_xpath('//tr[@height="19"]')
        for tr in tr_list:
            code = tr.find_element_by_xpath('./td[2]').text.strip()
            name = tr.find_element_by_xpath('./td[3]').text.strip()
            print(name,code)

    def run(self):
        self.get_incr_url()
        self.driver.quit()

if __name__ == '__main__':
  spider = GovSpider()
  spider.run()
```

3.4 切换frame

- **特点+方法**

```
【1】特点
    网页中嵌套了网页，先切换到iframe，然后再执行其他操作

【2】处理步骤
    2.1) 切换到要处理的Frame
    2.2) 在Frame中定位页面元素并进行操作
    2.3) 返回当前处理的Frame的上一级页面或主页面
```

```
【3】常用方法
    3.1) 切换到frame   -   driver.switch_to.frame(frame节点对象)
    3.2) 返回上一级    -   driver.switch_to.parent_frame()
    3.3) 返回主页面    -   driver.switch_to.default_content()

【4】使用说明
    4.1) 方法一：默认支持id和name属性值
        switch_to.frame(id属性值|name属性值)
    4.2) 方法二：
        a> 先找到frame节点 ： frame_node = driver.find_element_by_xpath('xxxx')
        b> 在切换到frame    : driver.switch_to.frame(frame_node)
```

- **示例1 - 登录豆瓣网**

```
"""
登录豆瓣网
"""
from selenium import webdriver
import time

# 打开豆瓣官网
driver = webdriver.Chrome()
driver.get('https://www.douban.com/')

# 切换到iframe子页面
login_frame = driver.find_element_by_xpath('//*[@id="anony-reg-new"]/div/div[1]/iframe')
driver.switch_to.frame(login_frame)

# 密码登录 + 用户名 + 密码 + 登录豆瓣
driver.find_element_by_xpath('/html/body/div[1]/div[1]/ul[1]/li[2]').click()
driver.find_element_by_xpath('//*[@id="username"]').send_keys('自己的用户名')
driver.find_element_by_xpath('//*[@id="password"]').send_keys('自己的密码')
driver.find_element_by_xpath('/html/body/div[1]/div[2]/div[1]/div[5]/a').click()
time.sleep(3)

# 点击我的豆瓣
driver.find_element_by_xpath('//*[@id="db-nav-sns"]/div/div/div[3]/ul/li[2]/a').click()
```

3.5 selenium总结

- **selenium+phantomjs|chrome|firefox小总结**

```
【1】设置无界面模式
    options = webdriver.ChromeOptions()
    options.add_argument('--headless')
    driver = webdriver.Chrome(excutable_path='/home/tarena/chromedriver',options=options)

【2】browser执行JS脚本
    driver.execute_script('window.scrollTo(0,document.body.scrollHeight)')
```

```
【3】鼠标操作
   from selenium.webdriver import ActionChains
   ActionChains(driver).move_to_element('node').perform()

【4】切换句柄 - switch_to.frame(handle)
   all_handles = driver.window_handles
   driver.switch_to.window(all_handles[1])

【5】iframe子页面
   driver.switch_to.frame(frame_node)
```

- **lxml中的xpath 和 selenium中的xpath的区别**

```
【1】lxml中的xpath用法 - 推荐自己手写
   eobj = etree.HTML(html)
   div_list = eobj.xpath('//div[@class="abc"]/div')
   item = {}
   for div in div_list:
       item['name'] = div.xpath('.//a/@href')[0]
       item['likes'] = div.xpath('.//a/text()')[0]

【2】selenium中的xpath用法 - 推荐copy - copy xpath
   # 此生铭记：selenium中的xpath表达式,千万不能加 /text() 和 /@属性名
   # 想获取文本：.text属性
   # 想获取属性值：.get_attribute('属性名')
   div_list = browser.find_elements_by_xpath('//div[@class="abc"]/div')
   item = {}
   for div in div_list:
       item['name'] = div.find_element_by_xpath('.//a').get_attribute('href')
       item['likes'] = div.find_element_by_xpath('.//a').text
```

4. 今日作业

```
【1】使用selenium+浏览器 获取有道翻译结果
   提示:1.text属性
        2.注意time.sleep()
【2】使用selenium+浏览器 登录网易163邮箱 : https://mail.163.com/
```